UNIVESITE SIDI MOHAMED BEN ABDELLAH

Ecole Nationale des Sciences Appliquées

Systèmes d'exploitation

Linux Ubuntu

- TP1 : Commandes de bases sous Linux.
- TP2 : Gestion des caractères et droits d'accès.
- TP3 : Programmation Shell

Préparé par : SELLAK MOUAD

INTRODUCTION :

Un système d'exploitation est un programme qui doit permettre aux utilisateurs d'utiliser les fonctionnalités d'un ordinateur, il doit aussi aider le programmeur à développer des logiciels de la façon la plus efficace possible, en plus un système d'exploitation est mis en route dès qu'on met en marche l'ordinateur, le système constitue donc une interface entre l'utilisateur et la machine.

Il existe différents types des systèmes d'exploitation telles que Windows, Mac, Unix … et chaque système se caractérise par un certain nombre de propriétés d'un autre.

On va s'intéresser au système d'exploitation Linux (Unix développé) qui porte les caractéristiques suivantes :

*-**Multi-utilisateurs** : plusieurs utilisateurs peuvent se connecter et travailler en même temps, chaque utilisateur a son propre environnement.*

*-**Multi-tâches** : un même utilisateur peut lancer plusieurs travaux simultanément.*

*-**Interactif** : il est possible de dialoguer avec l'ordinateur, et il y a aussi la possibilité de lancer des processus (taches) en arrière-plan et en mode différé (batch).*

On va travailler avec la version Ubuntu, qui travaille avec une interface simple, intuitive, et sécurisée, elle est proposée gratuitement sous forme d'image disque permettant de créer un live CD ou live USB afin de tester Ubuntu sans installation et sans modifier le système utilisé.

TP1 : Commandes de bases sous Linux

Exercice1 :

Commande	Description
man	*-Retourne le mode d'emploi de la commande s'il existe.* *-Syntaxe : man option commande* *-Ex : man man*
man -k	*-Liste tous les commandes associées au mot clé donné.* *-Syntaxe : man -k mot clé* *-Ex : man -k copy*
cd	*-Permet de déplacer dans le système des fichiers.* *-Syntaxe : cd non du répertoire* *-Ex :cd Desktop*
cd ..	*-Permet de se déplacer dans le répertoire parent.*
ls	*-Permet de lister le contenu du répertoire courant* *-Syntaxe : ls option fichier* *-Ex : ls*
ls -a	*-Lister le contenu de répertoire courant sans ignorer les -fichiers cachés (qui commencent par.)* *-Syntaxe : ls -a fichier* *-Ex :ls -a*
ls -l	*-Lister le contenu de répertoire courant avec plus de -détails tels que les droits d'accès, la date de la création…* *-Syntaxe : ls -l fichier* *- :ls -l*
ls -al	*-Exécute le rôle de ls -a et ls -l au même temps.*
pwd	*-Affiche le nom du répertoire courant* *-Syntaxe : pwd option* *-Ex : pwd*
mkdir	*-Permet de créer un nouveau répertoire.* *-Syntaxe : mkdir option repertoire* *-Ex : mkdir rep*
cp	*-Copier un fichier.* *-Syntaxe : cp option source de destination*

	-Ex : cp TP1 Documents
cp -R	*-Copie un répertoire d'une manière récursive.* *-Syntaxe : cp -R source de destination* *-Ex : cp -R TP1 TP*
mv	*-Permet de déplacer ou renommer un fichier.* *-Syntaxe : cp -R source de destination* *-Ex : cp -R TP1 TP*
rm	*-efface un fichier.* *-Syntaxe : rm option fichier* *-Ex : rm essai*
rm -R	*- efface récursivement un répertoire et son contenu.* *-Syntaxe : rm -R fichier* *-Ex : rm -R TP1*
who	*-Indique qui est connecté.* *-Syntaxe : who option fichier argument1 argument2…* *-Ex : who*
echo	*-Affiche une ligne de texte* *-Syntaxe : echo option "texte"* *-Ex : echo ''Bonjour''*
passwd	*-Modifie le mot de passe d'un utilisateur.* *-Syntaxe : passwd option login* *-Ex : passwd 0000*
history	*-Montre à l'utilisateur les dérnières commandes utilisées* *-Syntaxe : history nombre* *-Ex : history 5*
history -c	*-Réinitialise l'history.* *-Syntaxe : history -c* *-Ex : history -c*
touch	*-Permet de créer des fichiers.* *-Syntaxe : touch fichier* *-Ex : touch fich1*
cat	*-Concatène des fichiers et affiche le résultat dans le terminal.* *-Syntaxe : cat option fichier* *-Ex : cat essai*

Exercice2 :

Exercice 2 : (prise en main des commandes)

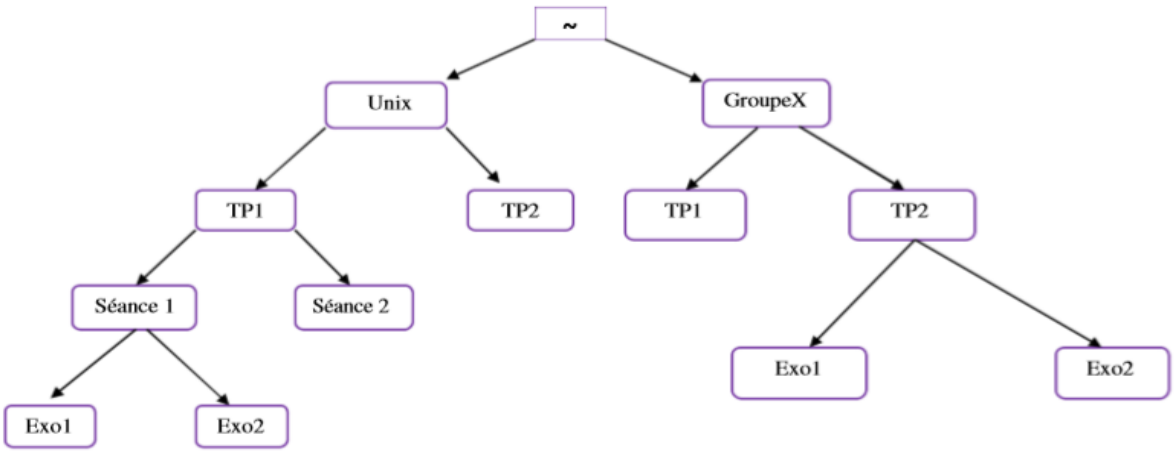
1. Tapez la commande **pwd**. Ce qui s'affiche est le nom absolu de votre répertoire personnel .Cela vous indique le chemin allant de la racine du système de fichiers à ce répertoire. Comment pouvez-vous revenir dans ce répertoire à partir de n'importe quel endroit dans le système de fichiers ?
2. Créez un répertoire Premier_repertoire (notez que l'on n'utilise ni le caractère" ", ni les accents).
3. Quelle commande pouvez-vous utiliser pour vérifier que le répertoire a bien été créé ?
4. Placez-vous dedans (grâce à la commande **cd**). Comme le nom est un peu long à taper, utilisez la touche **tab** pour le compléter.
5. Utilisez les flèches "haut" et "bas" pour faire défiler l'historique des commandes que vous avez déjà tapées et faites à nouveau afficher le nom absolu du répertoire où vous vous trouvez.

```
ubuntu@ubuntu:~$ pwd
/home/ubuntu
ubuntu@ubuntu:~$ cd ..
ubuntu@ubuntu:/home$ cd
ubuntu@ubuntu:~$ mkdir Premier_repertoire
ubuntu@ubuntu:~$ ls
Desktop     Music              Premier_repertoire  Videos
Documents   Pictures           Public              examples.desktop
Downloads   PremierRepertoire  Templates
ubuntu@ubuntu:~$ cd Premier_repertoire/
ubuntu@ubuntu:~/Premier_repertoire$ pwd
/home/ubuntu/Premier_repertoire
```

Exercice2 :

Exercice 3 : (manipulation des fichiers et répertoires)

1. Créer dans votre répertoire personnel les fichiers/répertoires selon l'arborescence suivante :



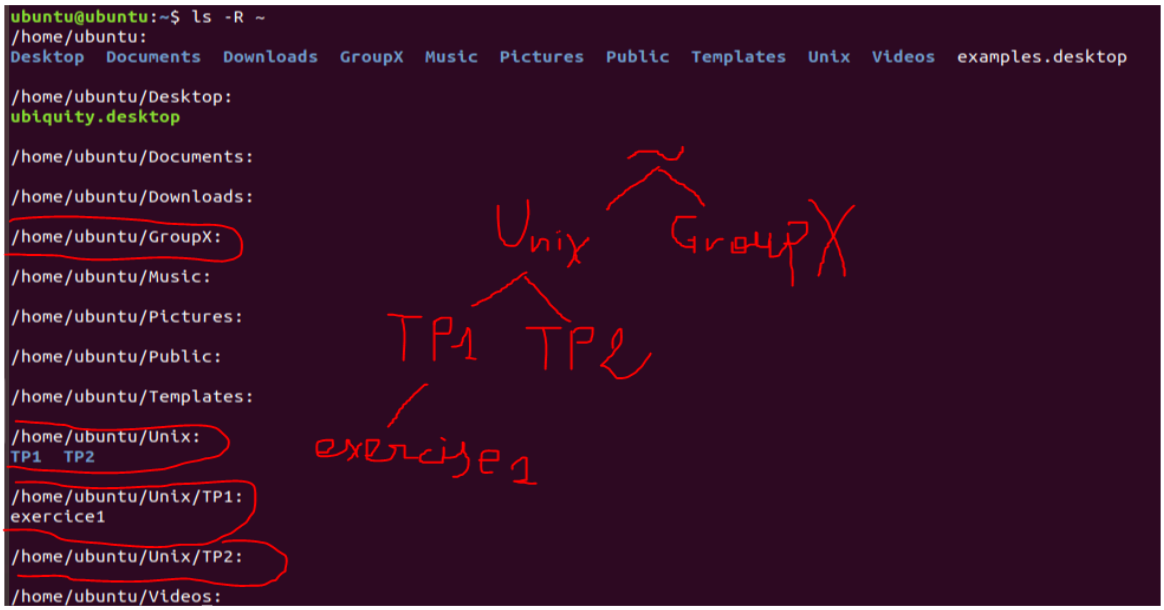
Rappelons que chaque question doit être répondue en une seule commande et le répertoire courant doit toujours être votre répertoire personnel, vous procédez de la façon suivante :

a) Créer deux répertoires Unix et Groupe X dans votre répertoire personnel.
b) Créer deux répertoires TP1 et TP2 dans le répertoire Unix.
c) Copier les deux répertoires TP1 et TP2 dans le répertoire GroupeX.
d) Créer deux répertoires seance1 et seance2 dans le répertoire TP1 de Unix.
e) Créer deux fichiers exo1 et exo2 dans le répertoire seance1.
f) Copier les deux fichiers exo1et exo2 dans le répertoireTP2de GroupeX.

```
ubuntu@ubuntu:~$ mkdir Unix GroupeX
ubuntu@ubuntu:~$ ls
Desktop    GroupeX  PremierRepertoire   Templates  examples.desktop
Documents  Music    Premier_repertoire  Unix
Downloads  Pictures Public              Videos
ubuntu@ubuntu:~$ cd Unix
ubuntu@ubuntu:~/Unix$ mkdir TP1 TP2
ubuntu@ubuntu:~/Unix$ cd
ubuntu@ubuntu:~$ cp -R Unix/TP1 Unix/TP2 GroupeX/
ubuntu@ubuntu:~$ ls Unix/
TP1  TP2
ubuntu@ubuntu:~$ ls GroupeX/
TP1  TP2
ubuntu@ubuntu:~$ cd Unix/TP1
ubuntu@ubuntu:~/Unix/TP1$ mkdir seance1 seance2
ubuntu@ubuntu:~/Unix/TP1$ ls
seance1  seance2
ubuntu@ubuntu:~/Unix/TP1$ cd seance1
ubuntu@ubuntu:~/Unix/TP1/seance1$ touch exo1 exo2
ubuntu@ubuntu:~/Unix/TP1/seance1$ ls
exo1  exo2
ubuntu@ubuntu:~/Unix/TP1/seance1$ cd
ubuntu@ubuntu:~$ cp -R Unix/TP1/seance1/exo1 Unix/TP1/seance1/exo2 GroupeX/TP2
ubuntu@ubuntu:~$ ls GroupeX/TP2
exo1  exo2
```

2. Placez-vous dans le répertoire GroupeX/TP1. Vérifiez que vous y êtes, puis, en une seule commande, placez-vous dans le répertoire Unix/TP2 en utilisant le chemin relatif du répertoire. Retournez dans GroupeX/TP1, mais cette fois-ci, en utilisant le chemin absolu. Enfin, revenez à la racine de votre répertoire personnel.
3. Supprimer le répertoireTP1 de GroupeX avec la commande **rmdir**.
4. Supprimer le répertoire TP2 de GroupeX avec la commande **rmdir**, pourquoi la commande a échoué. Supprimer en une commande le répertoireTP2 de GroupeX.
5. Déplacer le fichier exo1 dans le répertoireTP1 et supprimer les répertoires seance1 et seance2.
6. Renommer les fichiers exo1et exo2 en respectivement exercice1et exercice2.
7. Dessiner l'arborescence des fichiers contenus dans le répertoire personnel de manière théorique. Vérifier le à l'aide d'une commande.

```
ubuntu@ubuntu:~$ cd GroupeX/TP1
ubuntu@ubuntu:~/GroupeX/TP1$ cd /home/ubuntu/Unix/TP2
ubuntu@ubuntu:~/Unix/TP2$ cd ~/GroupeX/TP1
ubuntu@ubuntu:~/GroupeX/TP1$ cd
ubuntu@ubuntu:~$ rmdir GroupeX/TP1
ubuntu@ubuntu:~$ rmdir GroupeX/TP2
rmdir: failed to remove 'GroupeX/TP2': Directory not empty
ubuntu@ubuntu:~$ rm -R  GroupeX/TP2
ubuntu@ubuntu:~$ ls GroupeX/
ubuntu@ubuntu:~$ cp -R Unix/TP1/seance1/exo1 Unix/TP1
ubuntu@ubuntu:~$ ls Unix/TP1
exo1  seance1  seance2
ubuntu@ubuntu:~$ rm -R Unix/TP1/seance1 Unix/TP1/seance2
ubuntu@ubuntu:~$ ls Unix/TP1
exo1
ubuntu@ubuntu:~$ mv Unix/TP1/exo1 Unix/TP1/exercice1
ubuntu@ubuntu:~$ ls Unix/TP1
exercice1
```


ubuntu@ubuntu:~$ ls -R ~
/home/ubuntu:
Desktop Documents Downloads GroupX Music Pictures Public Templates Unix Videos examples.desktop
/home/ubuntu/Desktop:
ubiquity.desktop
/home/ubuntu/Documents:
/home/ubuntu/Downloads:
/home/ubuntu/GroupX:
/home/ubuntu/Music:
/home/ubuntu/Pictures:
/home/ubuntu/Public:
/home/ubuntu/Templates:
/home/ubuntu/Unix:
TP1 TP2
/home/ubuntu/Unix/TP1:
exercice1
/home/ubuntu/Unix/TP2:
/home/ubuntu/Videos:
~
Unix
GroupX
TP1
TP2
exercice1

TP2 : Gestion des caractères et droits d'accès

Exercice 1 : (méta-caractères)

1. Dans votre répertoire courant, créer en une commande les fichiers suivants : **3bananes 3poires 45pommes aride banane paname panorama Paris parti parties pomme programmer.**
2. Créer le répertoire **Dictionnaire** dans votre répertoire courant. Déplacer les fichiers précédemment créés dans le répertoire Dictionnaire.
3. A partir du répertoire Dictionnaire lister tous les fichiers :
   a) Se terminant par **e**.
   b) Commençant par **pa**.
   c) Contenant la chaîne **ar**.
   d) Commençant par un chiffre et ayant ensuite exactement quatre lettres minuscules.
   e) Commençant par un **chiffre** et contenant au moins trois lettres **minuscules** consécutives dont la troisième lettre de ce groupe est une **voyelle**.
   f) Commençant par un **chiffre** et contenant au moins trois lettres **minuscules** consécutives dont la troisième lettre de ce groupe n'est pas une **voyelle**.
   g) Contenant la chaîne **ar**, celle-ci n'étant pas suivie d'un **i**.
   h) Commençant par **p** ou **P.**
   i) Commençant par **p** ou **P**, ou finissant par un **s.**

   j) Commençant ni par **p**, ni par **P**.
   k) Ne terminant ni par **e**, ni par **s**.
   l) Dont l'avant dernier caractère est un **m** ou le dernier est un **e.**
   m) Dont le troisième caractère est une **voyelle**.
   n) Ne terminant pas par **me**.
   o) Ne terminant pas par **mes**.
   p) Ne terminant ni par **mes**, ni par **me**.

```
ubuntu@ubuntu:~$ mkdir TP2
ubuntu@ubuntu:~$ cd TP2
ubuntu@ubuntu:~/TP2$ touch 3bananes 3poires 45pommes aride banane paname panorama Paris parti parties programmer
ubuntu@ubuntu:~/TP2$ ls
3bananes  3poires  45pommes  Paris  aride  banane  paname  panorama  parti  parties  programmer
ubuntu@ubuntu:~/TP2$ mkdir Dictionnaire
ubuntu@ubuntu:~/TP2$ mv 3bananes 3poires 45pommes aride banane paname panorama Paris parti parties programmer Dictionnaire/
ubuntu@ubuntu:~/TP2$ ls
Dictionnaire
ubuntu@ubuntu:~/TP2$ ls Dictionnaire/
3bananes  3poires  45pommes  Paris  aride  banane  paname  panorama  parti  parties  programmer
ubuntu@ubuntu:~/TP2$ cd Dictionnaire/
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*e"
./paname
./banane
./aride
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "pa*"
./parties
./parti
./panorama
./paname
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*ar*"
./parties
./parti
./Paris
./aride
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[0-9][a-z][a-z][a-z][a-z]"
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[0-9][a-z][a-z][aouiey]*"
./3poires
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[0-9][a-z][a-z][!aouiey]*"
./3bananes
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*ar[!i]*"
./parties
./parti
```

```
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[pP]*"
./programmer
./parties
./parti
./Paris
./panorama
./paname
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[pP]*" -o -name "*s"
./programmer
./parties
./parti
./Paris
./panorama
./paname
./45pommes
./3poires
./3bananes
```

```
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[!P]*" -o -name "[!p]*"
.
./programmer
./parties
./parti
./Paris
./panorama
./paname
./banane
./aride
./45pommes
./3poires
./3bananes
```

```
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*[!e]" -a -name "*[!s]"
.
./programmer
./parti
./panorama
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "[!P]*" -a -name "[!p]*"
.
./banane
./aride
./45pommes
./3poires
./3bananes
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*[m]?" -o -name "*e"
./panorama
./paname
./banane
./aride
```

```
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "??[aueioy]*"
./programmer
./aride
./3poires
./3bananes
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*[!me]"
.
./programmer
./parties
./parti
./Paris
./panorama
./45pommes
./3poires
./3bananes
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*[!mes]"
.
./programmer
./parti
./panorama
ubuntu@ubuntu:~/TP2/Dictionnaire$ find -name "*[!mes]" -a -name "*[!me]"
.
./programmer
./parti
./panorama
```

Exercice 2 : (Compte utilisateurs)

1. Créer trois comptes utilisateurs, **mtrma**, **mtrmb**, and **mtrmc**.
2. Créer un nouveau groupe, **midterm**, avec **GID=5000**,
3. Ajouter chaque utilisateur, **mtrma**, **mtrmb**, et **mtrmc** dans le groupe **midterm**. Chaque groupe utilisateur doit rester dans son groupe.

```
ubuntu@ubuntu:~$ sudo useradd mtrma
ubuntu@ubuntu:~$ sudo useradd mtrmb
ubuntu@ubuntu:~$ sudo useradd mtrmc
ubuntu@ubuntu:~$ sudo groupadd midterm -g 5000
ubuntu@ubuntu:~$ sudo usermod -g midterm mtrma
ubuntu@ubuntu:~$ sudo usermod -g midterm mtrmb
ubuntu@ubuntu:~$ sudo usermod -g midterm mtrmc
ubuntu@ubuntu:~$ vi /etc/passwd
```

```
ubuntu:x:999:999:Live session user,,,:/home/ubuntu:/bin/bash
mtrma:x:1000:5000::/home/mtrma:/bin/sh
mtrmb:x:1001:5000::/home/mtrmb:/bin/sh
mtrmc:x:1002:5000::/home/mtrmc:/bin/sh
```

Exercice 3 : (droits d'accès)

1. Dans votre répertoire courant, créer un répertoire **essai_droit** . Quels sont les droits par défaut de ce répertoire ?
2. Quelles sont les commandes (en **notation symbolique** et en **notation octale**) pour lui donner les droits suivants :

	Propriétaire			Groupe			Autre		
	Lec	Ecri	Exé	Lec	Ecri	Exé	Lec	Ecri	Exé
Commande 1	Oui	Oui	Oui	Oui	Non	Oui	Non	Non	Oui
Commande 2	Oui	Non	Oui	Non	Oui	Non	Non	Non	Oui
Commande 3	Non	Oui	Non	Non	Non	Oui	Oui	Non	Non
Commande 4	Non	Non	Oui	Oui	Non	Oui	Non	Non	Non

3. Tester ces commandes et vérifier les droits d'accès obtenus à chaque étape.
4. Créer un fichier **droit** dans le répertoire **essai_droit**, quels sont ses droits par défaut ? En partant du répertoire courant, pour chaque commande de la question précédente, essayer d'accéder au répertoire **essai_droit** (commande **cd**), de lister le contenu du répertoire **essai_droit** (commande **ls**) et de modifier le fichier **droit** avec un éditeur quelconque (**emacs** ou **vi** par exemple).

	Propriétaire			Groupe			Autre			Notation symbolique	Notation octale
	Lec	Ecri	Exé	Lec r	Ecri w	Exé x	Lec 4	Ecri 2	Exé 1		
Commande 1	Oui	Oui	Oui	Oui	Non	Oui	Non	Non	Oui	chmod u=rwx,g=rx,o=x essai_droit	chmod 751 essai_droit
Commande 2	Oui	Non	Oui	Non	Oui	Non	Non	Non	Oui	chmod u=rx,g=w,o=x essai_droit	chmod 521 essai_droit
Commande 3	Non	Oui	Non	Non	Non	Oui	Oui	Non	Non	chmod u=w,g=x,o=r essai_droit	chmod 214 essai_droit
Commande 4	Non	Non	Oui	Oui	Non	Oui	Non	Non	Non	chmod u=x,g=rx,o=- essai_droit	chmod 150 essai_droit

Notation octale :

```
ubuntu@ubuntu:~/TP2$ mkdir essai_droit
ubuntu@ubuntu:~/TP2$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 40 Apr 21 10:04 essai_droit
ubuntu@ubuntu:~/TP2$ chmod 751 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
drwxr-x--x 2 ubuntu ubuntu 40 Apr 21 10:04 essai_droit
ubuntu@ubuntu:~/TP2$ chmod 521 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
dr-x-w---x 2 ubuntu ubuntu 40 Apr 21 10:04 essai_droit
ubuntu@ubuntu:~/TP2$ chmod 214 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
d-w---xr-- 2 ubuntu ubuntu 40 Apr 21 10:04 essai_droit
ubuntu@ubuntu:~/TP2$ chmod 150 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
d--xr-x--- 2 ubuntu ubuntu 40 Apr 21 10:04 essai_droit
```

Notation symbolique :

```
ubuntu@ubuntu:~/TP2$ mkdir essai_droit
ubuntu@ubuntu:~/TP2$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 40 Apr 21 10:08 essai_droit
ubuntu@ubuntu:~/TP2$ chmod o-r essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
drwxr-x--x 2 ubuntu ubuntu 40 Apr 21 10:08 essai_droit
ubuntu@ubuntu:~/TP2$ chmod u-w,g=w essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
dr-x-w---x 2 ubuntu ubuntu 40 Apr 21 10:08 essai_droit
ubuntu@ubuntu:~/TP2$ chmod u=w,g=x,o=r essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
d-w---xr-- 2 ubuntu ubuntu 40 Apr 21 10:08 essai_droit
ubuntu@ubuntu:~/TP2$ chmod u=x,g+x,o-r essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
d--x--x--- 2 ubuntu ubuntu 40 Apr 21 10:08 essai_droit
ubuntu@ubuntu:~/TP2$ chmod u=x,g+r,o-r essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
d--xr-x--- 2 ubuntu ubuntu 40 Apr 21 10:08 essai_droit
```

```
ubuntu@ubuntu:~/TP2/essai_droit$ touch droit
ubuntu@ubuntu:~/TP2/essai_droit$ ls
droit
ubuntu@ubuntu:~/TP2/essai_droit$ ls -l
total 0
-rw-r--r-- 1 ubuntu ubuntu 0 Apr 21 10:42 droit
ubuntu@ubuntu:~/TP2/essai_droit$ cd ..
ubuntu@ubuntu:~/TP2$ chmod 751 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
drwxr-x--x 2 ubuntu ubuntu 60 Apr 21 10:42 essai_droit
ubuntu@ubuntu:~/TP2$ cd essai_droit/
ubuntu@ubuntu:~/TP2/essai_droit$ ls -l
total 0
-rw-r--r-- 1 ubuntu ubuntu 0 Apr 21 10:42 droit
ubuntu@ubuntu:~/TP2/essai_droit$ vi droit
ubuntu@ubuntu:~/TP2/essai_droit$ cat droit
C
ubuntu@ubuntu:~/TP2/essai_droit$ ls -l
total 4
-rw-r--r-- 1 ubuntu ubuntu 2 Apr 21 10:49 droit
ubuntu@ubuntu:~/TP2/essai_droit$ cd ..
ubuntu@ubuntu:~/TP2$ chmod 521 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
dr-x-w---x 2 ubuntu ubuntu 60 Apr 21 10:49 essai_droit
ubuntu@ubuntu:~/TP2$ cd essai_droit/
ubuntu@ubuntu:~/TP2/essai_droit$ vi droit
ubuntu@ubuntu:~/TP2/essai_droit$ cat droit
Cje suis dans le fichier droit
ubuntu@ubuntu:~/TP2/essai_droit$ ls -l
total 4
-rw-r--r-- 1 ubuntu ubuntu 31 Apr 21 10:52 droit
```

```
ubuntu@ubuntu:~/TP2/essai_droit$ cd ..
ubuntu@ubuntu:~/TP2$ chmod 214 essai_droit/
ubuntu@ubuntu:~/TP2$ ls -l
total 0
d-w---xr-- 2 ubuntu ubuntu 60 Apr 21 10:49 essai_droit
ubuntu@ubuntu:~/TP2$ cd essai_droit/
bash: cd: essai_droit/: Permission denied
ubuntu@ubuntu:~/TP2$ chmod 150 essai_droit/
ubuntu@ubuntu:~/TP2$ cd essai_droit/
ubuntu@ubuntu:~/TP2/essai_droit$ vi droit
ubuntu@ubuntu:~/TP2/essai_droit$ cat droit
Cette fois j'qjoute cette ligne
ubuntu@ubuntu:~/TP2/essai_droit$ ls -l
ls: cannot open directory '.': Permission denied
```

TP3 : Programmation Shell

```
ubuntu@ubuntu:~$ gedit PremierScriptShell.sh
ubuntu@ubuntu:~$ chmod 777 PremierScriptShell.sh
ubuntu@ubuntu:~$ bash ./PremierScriptShell.sh
Hello world!
ubuntu@ubuntu:~$ sh -x PremierScriptShell.sh
+ echo Hello world!
Hello world!
ubuntu@ubuntu:~$ cat PremierScriptShell.sh
#!/bin/bash

echo "Hello world! "

ubuntu@ubuntu:~$ gedit hello.c
ubuntu@ubuntu:~$ cat hello.c
#include <stdio.h>
int main()
{
printf("Hello world ! \n ");
return 0;
}
ubuntu@ubuntu:~$ chmod +x hello.c
ubuntu@ubuntu:~$ gcc -o hello hello.c
ubuntu@ubuntu:~$ ./hello
Hello world !
```

Exercice 1 :

Ecrire un script Shell qui vous demande d'entrer un nom d'utilisateur et un mot de passe ; si le nom d'utilisateur est ''admin'' et le mot de passe est ''mot_admin'' le résultat retournée sera utilisateur valide sinon il retourne utilisateur invalide .

```
#!/bin/bash

echo "Entrez le nom de votre utilisateur: "
 read utilisateur
echo "Entrez le mot de passe :"
 read paswd

if [[ "$utilisateur" = "admin" && "$paswd" = "mot_admin" ]]
   then
     echo "Utilisateur valide"
else
    echo "Utilisateur invalide"
fi
```

```
ubuntu@ubuntu:~$ gedit Utilisateur.sh
ubuntu@ubuntu:~$ bash ./Utilisateur.sh
Entrez le nom de votre utilisateur:
admin
Entrez le mot de passe :
mot_admin
Utilisateur valide
ubuntu@ubuntu:~$ bash ./Utilisateur.sh
Entrez le nom de votre utilisateur:
aaaa
Entrez le mot de passe :
bbbb
Utilisateur invalide
```

Exercice 2 :

Ecrire un script Shell qui permet d'entrer deux nombre au clavier, et de les comparer, et de calculer l'addition de ses deux nombres.

```
#!/bin/bash
echo "sasir le premier nombre:"
 read n1

echo "sasir le deuxieme nombre:"
 read n2

if [ "$n1" -le "$n2" ]
 then
   echo "$n1 est inferieur a $n2"
else
    echo "$n1 est superieur  a $n2"
fi
somme=0
  let somme=n1+n2
echo " la somme est $somme "
```

```
ubuntu@ubuntu:~$ bash ./Operation.sh
sasir le premier nombre:
23
sasir le deuxieme nombre:
7
23 est superieur  a 7
 la somme est 30
```

Exercice 3 :

Ecrire un scripte qui tri un tableau

```
#!/bin/bash

 echo "saisir la taille de votre tableau: "
  read -r taille

 echo "saisir vos elements:"
for ((i=0;i<taille;i++)) ; do
  read -r tab[i]
done

 echo "Avant le tri: "
for ((i=0;i<taille;i++)) ; do
    echo "${tab[$i]}"
 done

for ((i=0;i<taille;i++)) ; do
for ((j=$i+1;j<taille;j++)) ; do
      if [ "${tab[$j]}" -le "${tab[$i]}" ]
       then
          tmp="${tab[$i]}"
          tab[$i]="${tab[$j]}"
          tab[$j]="$tmp"
      fi
  done
done

 echo "Apres le tri"
for ((i=0;i<taille;i++)) ; do
 echo "${tab[$i]}"
 done
```

```
ubuntu@ubuntu:~$ gedit Tri_a_bulles.sh
ubuntu@ubuntu:~$ chmod +x Tri_a_bulles.sh
ubuntu@ubuntu:~$ bash ./Tri_a_bulles.sh
saisir la taille de votre tableau:
4
saisir vos elements:
7
6
9
1
Avant le tri:
7
6
9
1
Apres le tri
1
6
7
9
```

Exercice 4 :

Ecrire un script Shell qui liste toute les fichiers dans votre répertoire personnel.

```
#!/bin/bash

printf "%s \n" *
```

```
ubuntu@ubuntu:~$ ls
Desktop  Documents  Downloads  Lister1.sh  Music  Pictures  Public  TP  Templates  Videos  examples.desktop  lister.sh
ubuntu@ubuntu:~$ gedit lister.sh
ubuntu@ubuntu:~$ chmod +x lister.sh
ubuntu@ubuntu:~$ bash lister.sh
Desktop
Documents
Downloads
Lister1.sh
Music
Pictures
Public
TP
Templates
Videos
examples.desktop
lister.sh
```

Autre méthode :

```
#!/bin/bash

files=(*)
for file in "${files[@]}" ; do
 echo "$file"
done
```

```
ubuntu@ubuntu:~$ gedit Lister1.sh
ubuntu@ubuntu:~$ chmod +x Lister1.sh
ubuntu@ubuntu:~$ bash Lister1.sh
Desktop
Documents
Downloads
Lister1.sh
Music
Pictures
Public
TP
Templates
Videos
examples.desktop
lister.sh
ubuntu@ubuntu:~$ ls
Desktop  Documents  Downloads  Lister1.sh  Music  Pictures  Public  TP  Templates  Videos  examples.desktop  lister.sh
```

Exercice 5 :

Ecrire un script qui permet de retourné un mot quelconque dans une phrase.

```
#!/bin/bash

printf "Combien de mot vous voulez inclure dans votre phrase ?"
 read n
printf "Entrez votre phrase en respetant le nombre de mots : \n"

for (( i=1; i<=$n ;i++ )) ; do
 read -r phrase[i]
done

 echo "Entrez un entier compris entre 0 et $n :"
    read j
 echo "Le mot retourne est : ${phrase[$j]}"
```

```
ubuntu@ubuntu:~$ gedit Routourner.sh
ubuntu@ubuntu:~$ chmod +x Routourner.sh
ubuntu@ubuntu:~$ bash Routourner.sh
Combien de mot vous voulez inclure dans votre phrase ?4
Entrez votre phrase en respetant le nombre de mots :
je
suis
mouad
sellak
Entrez un entier compris entre 0 et 4 :
3
Le mot retourne est : mouad
ubuntu@ubuntu:~$ bash Routourner.sh
Combien de mot vous voulez inclure dans votre phrase ?4
Entrez votre phrase en respetant le nombre de mots :
je
suis
un
etudiant
Entrez un entier compris entre 0 et 4 :
4
Le mot retourne est : etudiant
```

Exercice 6 :

Ecrire un script qui demande d'entrer au clavier un nom d'un répertoire et qui vérifier est-ce-que ce répertoire existe-il si oui ; fin de programme sinon il va le créé

```
#!/bin/bash
echo "Entrez un nom d'un repertoire:"
 read rep
if [ -r "$rep" ] ; then
   echo "Ce repertpoire existe "
else
 mkdir $rep
fi
```

```
ubuntu@ubuntu:~$ mkdir RepTest
ubuntu@ubuntu:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  RepTest  Repertoire.sh  Templates  Videos  examples.desktop  tmp
ubuntu@ubuntu:~$ gedit Repertoire.sh
ubuntu@ubuntu:~$ chmod +x Repertoire.sh
ubuntu@ubuntu:~$ bash ./Repertoire.sh
Entrez un nom d'un repertoire:
RepTest
Ce repertpoire existe
ubuntu@ubuntu:~$ bash ./Repertoire.sh
Entrez un nom d'un repertoire:
NewRep
ubuntu@ubuntu:~$ ls
Desktop  Documents  Downloads  Music  NewRep  Pictures  Public  RepTest  Repertoire.sh  Templates  Videos  examples.desktop  tmp
ubuntu@ubuntu:~$
```

Exercice 7 :

Ecrire un script qui contient une fonction qui calcule la surface d'un rectangle

```
#!/bin/bash

Calculer_la_surface()
{
echo "Entrez la longueur de votre rectengle:"
 read long
echo "Entrez la largeur de votre rectangle:"
 read larg
let surf=long*larg
echo "La surface de votre rectangle est : $surf "
}
#Appel de la fonction
Calculer_la_surface
```

```
ubuntu@ubuntu:~$ gedit Surface.sh
ubuntu@ubuntu:~$ chmod +x Surface.sh
ubuntu@ubuntu:~$ bash ./Surface.sh
Entrez la longueur de votre rectengle:
6
Entrez la largeur de votre rectangle:
4
La surface de votre rectangle est : 24
```

Exercice 8 :

Créer un script qui prend un nombre en saisie et l'élève à sa propre puissance

```
#!/bin/bash

echo "Entrez un entier :"
 read nbr
res=1
for((i=0;i<$nbr;i++)) ; do
 let res=res*nbr
done

echo "$nbr^$nbr=$res"
```

```
ubuntu@ubuntu:~$ gedit Puissance.sh
ubuntu@ubuntu:~$ chmod +x Puissance.sh
ubuntu@ubuntu:~$ bash ./Puissance.sh
Entrez un entier :
3
3^3=27
```

Exercice 9

Ecrire un programme en C qui affiche un triangle des étoiles et puis compiler et exécuter le programme a l'aide du compilateur GCC

```
ubuntu@ubuntu:~$ gedit tringle_etoles.c
ubuntu@ubuntu:~$ chmod 777 tringle_etoles.c
ubuntu@ubuntu:~$ cat tringle_etoles.c
#include<stdio.h>

int main()
{
int i,j,n;
printf("Combien vous voulez de lignes : ");
scanf("%d",&n);
for(i=1;i<n;i++){
  for(j=0;j<i;j++){
   printf("*");
   }
 printf("\n");
 }
return 0;
}
ubuntu@ubuntu:~$ gcc -o tringle_etoles tringle_etoles.c
ubuntu@ubuntu:~$ ./tringle_etoles
Combien vous voulez de lignes : 6
*
**
***
****
*****
```

Exercice 10

Ecrire un programme en C qui calcule la multiplication de deux matrices carrées compiler le sur le terminale a l 'aide du compilateur GCC.

```
ubuntu@ubuntu:~$ gedit Produit_Matriciel.c
ubuntu@ubuntu:~$ cat Produit_Matriciel.c
#include<stdio.h>
int main()
{
int i,j,k,lin=3,col;
int A[lin][col] , B[lin][col] , C[lin][col] ;
col=lin;
printf(" \n Remlissage de la matrice A:\n ");
for(i=0;i<lin;i++){
  for(j=0;j<col;j++){
  printf("A[%d][%d]=",i,j);
  scanf("%d",&A[i][j]);
 }
}
printf(" \n Remlissage de la matrice B:\n ");
for(i=0;i<lin;i++){
  for(j=0;j<col;j++){
  printf("B[%d][%d]=",i,j);
  scanf("%d",&B[i][j]);
 }
}
printf(" \n Affichage de la matrice A:\n ");
for(i=0;i<lin;i++){
  for(j=0;j<col;j++)
  printf("%5d",A[i][j]);
  printf("\n");
}
printf(" \n Affichage de la matrice B:\n ");
for(i=0;i<lin;i++){
  for(j=0;j<col;j++)
  printf("%5d",B[i][j]);
  printf("\n");
}
```

```
for(i=0;i<lin;i++) {
  for(j=0;j<col;j++) {
      C[i][j]=0;
    for(k=0;k<col;k++){
            C[i][j]=C[i][j] + A[i][k]*B[k][j];
          }
    }
}
printf(" \n Affichage de la matrice produit C:\n ");
for(i=0;i<lin;i++){
  for(j=0;j<col;j++)
  printf("%5d",C[i][j]);
  printf("\n");
}
return 0;
}
ubuntu@ubuntu:~$ chmod 777 Produit_Matriciel.c
ubuntu@ubuntu:~$ gcc -o Produit_Matriciel Produit_Matriciel.c
ubuntu@ubuntu:~$ ./Produit_Matriciel
```

```
 Remlissage de la matrice A:
 A[0][0]=1
A[0][1]=1
A[0][2]=1
A[1][0]=1
A[1][1]=1
A[1][2]=1
A[2][0]=1
A[2][1]=1
A[2][2]=1

 Remlissage de la matrice B:
 B[0][0]=2
B[0][1]=2
B[0][2]=2
B[1][0]=2
B[1][1]=2
B[1][2]=2
B[2][0]=2
B[2][1]=2
B[2][2]=2

 Affichage de la matrice A:
     1    1    1
    1    1    1
    1    1    1

 Affichage de la matrice B:
     2    2    2
    2    2    2
    2    2    2

 Affichage de la matrice produit C:
     6    6    6
    6    6    6
    6    6    6
```

Exercice 11 :

Ecrire un programme C qui calcule le factoriel d'un entier.

```
ubuntu@ubuntu:~$ gedit factoriel.c
ubuntu@ubuntu:~$ cat factoriel.c
#include<stdio.h>
int main()
{
int i,n,f=1;
printf("Entrez un nombre : ");
scanf("%d",&n);
if (n==0)
printf(" %d!=1",n);
else{
  for(i=n;i>0;i--)
   f=f*i;
}
printf("\n %d!=%d",n,f);
}
ubuntu@ubuntu:~$ chmod 777 factoriel.c
ubuntu@ubuntu:~$ gcc -o factoriel factoriel.c
ubuntu@ubuntu:~$ ./factoriel
Entrez un nombre : 4

 4!=24ubuntu@ubuntu:~$ ls
```

CONCLUSION :

*Force et d'avancer qu'on a appris pas mal de choses dans le fameux système d'exploitation Linux, notamment les commandes de bases telles que **cd**, **pwd**, **ls**, **mkdir**, **rmdir**, **cat**…, en plus on a bien compris la façon de chercher des fichiers dans un répertoire donné en utilisant la commande **find** et évoquant les caractères spéciaux : * et ? d'autre part on a étudié un peu la gestion des utilisateurs et groupes ainsi que l'assimilation des droits d'accès d'un fichier, sans oublier qu'on a bien sait programmer et créer des scripts dans le fameux éditeur **gedit**, que ce soit des programmes en langage C ou des scripts en langages **bash**.*